**Правила работы над проектом**

**1. Система контроля версий**

Ведение работы:

* Использование Git для контроля версий и совместной работы над кодом.
* Все изменения в коде начинаются с создания новой ветки от основной ветки **main**.

Создание веток:

* Ветки создаются для каждой новой задачи или функционала (**feature**), исправления ошибок (**bugfix**) и подготовки релизов (**release**).
* **Именование веток:** следует использовать четкую схему наименования:
  + Для нового функционала: **feature/описание-функционала**, например, **feature/add-class-schedule**.
  + Для исправления ошибок: **bugfix/описание-ошибки**, например, **bugfix/correct-timezone-issue**.
  + Для релизов: **release/номер-версии**, например, **release/1.0.0**.

Code Review (Код-ревью):

* После завершения работы над задачей разработчик создает Pull Request (PR) в основную ветку.
* Каждый PR должен проходить код-ревью минимум одним другим разработчиком команды для обеспечения качества кода и соответствия стандартам проекта.
* В процессе ревью могут быть запрошены дополнительные изменения для улучшения кода или исправления ошибок.

Подтверждение Pull Requests (PR):

* PR может быть смерджен с основной веткой только после успешного код-ревью и исправления всех замечаний.
* Для подтверждения изменений используются автоматические тесты и проверки кода, если таковые имеются в проекте.
* Проектный менеджер или ведущий разработчик отвечает за финальное подтверждение и слияние PR.

Работа с типами правок:

* **Запрос на изменение (новый функционал, feature):** Каждый новый функционал начинается с создания ветки **feature**, разработки в ней, последующего PR и код-ревью.
* **Релиз:** Релизы готовятся в ветках **release**, где собираются все необходимые изменения и новый функционал для следующей версии продукта. После тщательного тестирования и код-ревью ветка **release** сливается с **main** и помечается тегом версии.
* **Багфикс:** Исправления ошибок производятся в ветках **bugfix**, отдельно для каждой ошибки. После исправления создается PR, проводится код-ревью, и после одобрения изменения интегрируются в основную ветку.

**2. Процесс код-ревью и подтверждение PR.**

* **Code Review** проводится с целью обеспечения высокого качества кода, соответствия стандартам разработки и обмена знаниями между членами команды.
* Ревьюеры оценивают чистоту кода, логику работы новой функциональности или исправления и проверяют соответствие изменений задаче.
* Ревьюер может запросить изменения в коде, если это необходимо для улучшения качества или исправления ошибок. Разработчик вносит эти изменения и повторно отправляет код на ревью.
* После того как ревьюеры удовлетворены качеством кода и выполнением всех требований, PR подтверждается и сливается с основной веткой.

**3. Утверждение и слияние изменений.**

* Перед слиянием PR в основную ветку необходимо убедиться, что все автоматические тесты пройдены успешно.
* Важно также проверить, что изменения корректно работают во всех предполагаемых условиях, особенно если проект использует CI/CD для автоматического развертывания.
* Окончательное решение о слиянии PR принимает ведущий разработчик или проектный менеджер, основываясь на отзывах ревьюеров и результатах автоматических тестов.

**4. Документация изменений.**

* Важным аспектом работы является документирование всех изменений в коде и функциональности. Это включает в себя комментарии в коде, обновление технической документации и руководств пользователя.
* Для каждого PR должно быть четкое описание сделанных изменений, причин для этих изменений и, при необходимости, ссылок на соответствующие задачи или ошибки.